Journal of Engineering Science and Technology
Vol. 15, No. 3 (2020) 1855 - 1867

ENGLISH NUMBERS RECOGNITION BASED ON SIGN LANGUAGE USING LINE-SLOPE FEATURES AND PSO-DBN OPTIMIZATION METHOD

SUHAILA N. MOHAMMED*, HUDA M. RADA

Department of Computer Science, College of Science, University of Baghdad, Iraq
*Corresponding Author: suhailan.mo@sc.uobaghdad.edu.iq

Abstract

Sign Language is considered as the primary method that used by dumb and deaf people for communication. The need for a computer based system that has the ability for recognizing these signs are imperious for the dumb community. However; despite researchers have been attempted to find a solution to this problem in the previous few years but the results are still not good enough. In this paper, a system for identifying the number involved in hand gesture is introduced. Hue Saturation Value (HSV) color model is used to allocate hand region. Discriminated features are generated by finding the slope of the line that connects any two points in hand region. A combination of particle swarm optimization and deep belief network is then used to find the optimal feature subset from the generated slope features. National University of Singapore (NUS) hand posture public dataset is used for system evaluation and the achieved accuracy is 99.58% when the number of blocks is set to 7.

Keywords: Deep Belief Network (DBN), HSV color model, Line-slope features, Particle Swarm Optimization (PSO), Sign language.

1. Introduction

In our life, speech plays a vital role for carrying information from one person to another. But, the communication between normal people and people who are deaf and dumb is very difficult. Sign language is the only one way to communicate with deaf people [1]. A sign language is a language which based on using gestures instead of speech to transmit information by changing the shape of hand, hand orientation and movement, arms or body and lip-patterns [2]. Sign language recognition systems are needed during medical and legal appointments, educational and training sessions [3, 4].

Different works attempted to build systems to interpret hand sign. Cooper et al. [5] presented sub-units which are generated from both 2-D and 3-D tracking data of hand movements. These sub-units are then integrated using Markov models to identify the temporal changes between the sub-units. Pramada et al. [6] introduced an algorithm to identify the number of fingers which are opened during expressing an alphabet using sign language. Coordinates of the captured finger image is calculated and then stored into a database of templates. The coordinates of newly unknown alphabet are compared with coordinates stored in the database. If a match found then the alphabet is converted into text and audio form. Karthikeyan and Muthulakshmi [7] implemented a technique for recognizing the English letters using sign language. 20 highest discrete cosine transformation coefficients are extracted from each finger spelling and then given as input to support vector machine model. Kiranalli and Gengaje [8] proposed an algorithm for recognizing number from 0 to 9 included in hand gesture based on boundary tracing and fingertip detection.

Machine learning and optimization algorithms are gained popularity in sign language recognition. Kaluri and Reddy [9] used an improved version of genetic algorithm to extract features with sign gesture recognition ability and then used support vector machine for classification. Kaluri and Reddy [10] used genetic algorithm with fuzzy classifier to find out the optimal rules generated by fuzzy classifier. Skeleton joint information (augmented with active difference used by signatures) is populated into hidden Markov models by Kaluri and Pradeep [11] to achieve gesture segmentation and recognition.

Many works are performed with NUS hand posture dataset as evaluation material. Nguyen et al. [12] used kernel descriptors and achieved accuracy equal to 97.3%. Vishwakarma [13] extracted shape and texture features which gave accuracy of 94.6%. Ji et al. [14] proposed Binary Edge HOG Block (BEHB) features and achieved accuracy equal to 97.72%. Lahiani and Neji [15] generated HOG-LBP features and the final accuracy was 92.00%. Different accuracies achieved by the authors of [12-15] due to the variation in the discrimination ability of the extracted features in their proposed methods.

Despite different works are done on sign language recognition, but it is still difficult to determine which features can work effectively in identifying the number conveyed within the hand sign. In this paper, we propose a system to recognize English numbers based on hand sign language in which line-slope features are generated from hand region and fed into Particle Swarm Optimization – Deep Belief Network (PSO-DBN) for best feature subset selection. The main contribution of the proposed work is in the extraction and selection the optimal feature set for recognizing the number conveyed within the hand region. The structure of remaining paper as follows: Theoretical backgrounds of methods used in this work are demonstrated in Section2. The proposed methodology is described in details in Section 3. The experimental analysis and achieved results are shown in Section4. Finally, work conclusion and ideas for future work are presented in Section5.

2. Theoretical Background

2.1. Particle swarm optimization

Eberhart and Kennedy introduced Particle Swarm Optimization (PSO) algorithm in 1995. PSO composed of a population of individuals; each individual represents a potential solution which called a "particle". The continuous PSO algorithm can described as [16]:

- N particles represent the population.
- The position vector of i^{th} particle is $x_i=(x_{i1},x_{i2},\ldots,x_{iD})^T$.
- The velocity vector of i^{th} particle is $v_i= (v_{i1},v_{i2},\ldots,v_{iD})^T$.
- The fitness value of i^{th} particle is described as $f_i=$ Fitness(x_i).
- The best local position vector in the population is represented by $P_l= (P_{l1}, P_{l2},\ldots,P_{lD})^T$.
- The best forever global position vector in the population is $P_g= (P_{g1}, P_{g2},\ldots,P_{gD})^T$.
- Each particle adjusts its speed dynamically according to population flying experience, and moves to the best position based on values of P_l and P_g. Each particle updates its speed and position according to Eqs. (1) and (2) [16].

$$v_i(t+1) = wv_i(t) + c_1r_1[P_l(t) - x_i(t)] + c_2r_2\left[P_g(t) - x_i(t)\right] \tag{1}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{2}$$

where, t is iteration time, w is "*inertia facto*r" used to ensure that the particle will continue to fly in its previous direction, r_1, r_2 are random numbers between 0 and 1, and c_1, c_2 are nonnegative constants, called "learning factors" that used to adjust each iteration step length.

2.2. Deep belief network (DBN)

The recent advances in the learning of deep neural networks make it possible to overcome the vanishing gradient problem existing on previous neural network models. This problem has been solved using a pre-training step, where deep belief networks (DBNs) are formed by the stacked Restricted Boltzmann Machines (RBMs) that perform unsupervised learning. Once a pre-training step is done, network weights are further fine-tuned by propagation the error backward, while the network is treated as a feed-forward net [17]. DBNs are probabilistic graphical models which have multiple hidden layers. RBM is a Boltzmann machine which is restricted to have only one hidden layer and one visible layer and also have no visible-visible and hidden-hidden connections [18, 19]. The architecture model of DBNs is shown in Fig. 1.

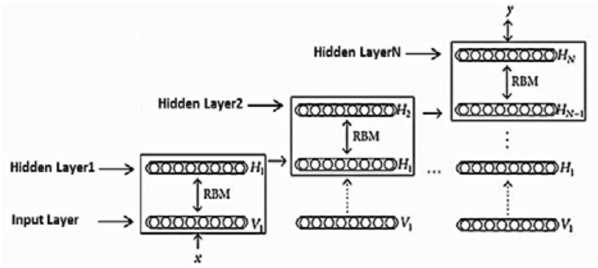


Fig. 1. Architecture of deep belief network (DBN).

3. The Proposed Sign Number Recognition System

The work flow of the proposed system involves four main stages (as shown in Fig. 2): (1) hand region allocation, (2) feature extraction, (3) optimal feature subset selection, and (4) classification stage.

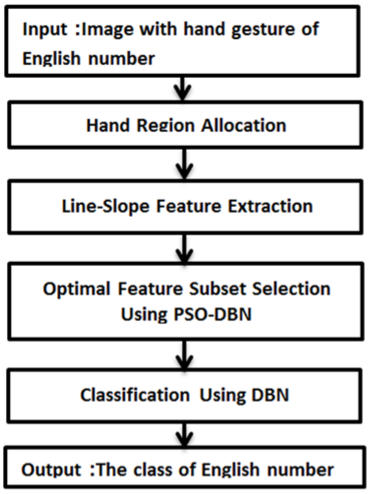


Fig. 2. Basic system design.

In hand region allocation stage, the hand region is defined and isolated. The main purpose of hand region allocation is to eliminate irrelevant background region and focus only on the hand area where the hand ROIs can be found. In feature extraction stage, the feature vector that is used to distinguish one number from another is extracted. Feature extraction stage involves points of interest extraction and line-slope feature generation. To reduce the number of generated features and select only strong ones, PSO-DBN method is employed in feature selection stage. Finally, the selected feature subset is fed into DBN to recognize the class number of the tested hand gesture image.

3.1. Hand region allocation

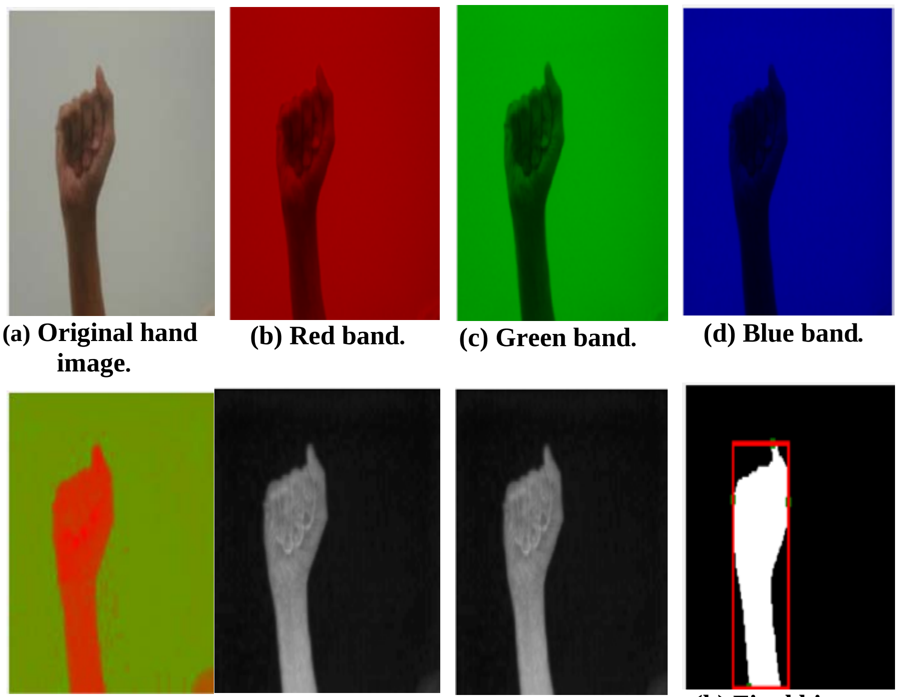
This stage consists of four main steps: Red, Green, Blue (RGB) color decomposition, RGB to(Hue, Saturation, Value (HSV) color model conversion, binarization and hand region isolation step. In RGB color decomposition step, the color of each pixel in the image is decomposed into its basic bands (i.e., red, green, blue). After RGB bands' extraction, the image is then converted into HSV color model. HSV is another representation for points in an RGB color model in which the color represented as a cylindrical-coordinate. The representation reorders the geometry of RGB in more intuitive and perceptually relevant manner. Equations (3), (4) and (5) are used to convert RGB color model to HSV color model [20].

$$H = \begin{cases} cos^{-1}\left(\frac{0.5[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right) & if\ B \leq G \\ 2\pi - cos^{-1}\left(\frac{0.5[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right) & if\ B > G \end{cases} \tag{3}$$

$$S = 1 - 3Min\ (R, G, B) \tag{4}$$

$$V = (R + G + B)/(3 \times 255) \tag{5}$$

The goal of binarization step is to segment the hand image into two regions: foreground (hand region) with white color and background with black color. The HSV color model is used since the saturation value resembles various tints of grey in the color space and can easily binariazed without the need for adaptive threshold value selection. Hand region separation step is done by opening search on the four directions of the binary image. The search stops when the first white pixel is hit for each direction. Figure 3 demonstrates the results of each step within hand region allocation stage.



(a) Original hand image. **(b) Red band.** **(c) Green band.** **(d) Blue band.**

(e) Hue band. **(f) Saturation band.** **(g) Value band.** **(h) Final binary image.**

Fig. 3. Results of hand region allocation stage.

3.2. Feature extraction stage

The feature vector used for recognition task is generated as follows:

Step1: The hand region is divided into *N* vertical blocks.

Step2: For each block, the coordinates of the white pixel laid in the max position within *y*-axis is recorded (i.e., saving its' *x*, *y* coordinates). That's mean there is a number of control points equal to the number of blocks. The width of each block depends on the number of blocks, as the latter increase the former will decrease and in effect reduce the number of white pixels in each block. Figure 4 shows locations of control points on the hand region.

Fig. 4. Control points' locations on the hand region.

Step3: The slopes of the lines that connect any two control points (without duplication) are calculated. The slope of the line can be measured using Eq. (6) [21].

$$Slope\ (p1, p2) = \frac{y_2 - y_1}{x_2 - x_1} \tag{6}$$

where: $p1$ and $p2$ are points, (x_1,y_1) and (x_2,y_2) are point coordinates, respectively. The length of the feature vector is equal to $\frac{M*(M-1)}{2}$ where M is the number of control points. As the number of M increases the number of extracted features will increase and in turn improve the accuracy. However; this will distress the task of feature selection. Thus, the best number of extracted features must be determined by experiments

Step4: The generated features may be of different scales for different hand images due to the effect of different camera zooming. To solve this problem, normalization process is applied on the generated features. Equation (7) is used for feature normalization [22].

$$V_n = \left(\frac{V_o - Mino}{Maxo - Mino}\right)(Maxn - Minn) + Min \tag{7}$$

where V_o and V_n are the old and new feature values, respectively, $Mino$ and $Maxo$ are the old range confines and $Minn$ and $Maxn$ are the new range confines.

3.3. Optimal feature-subset selection

With the recent advances in machine learning techniques, the belief that "the more the attributes, the better the performance" has not been acceptable; because as the number of attributes increases, the cost of computation also increases. Therefore, feature selection techniques have been earning popularity in the field of data mining.

PSO-DBN method is employed in this work for feature selection task in which PSO algorithm is used along with DBN as fitness function to evaluate the accuracy achieved with the selected feature subset during optimization process. Out of ($\frac{M*(M-1)}{2}$) generated line-slope features, only the best T features that give the best accuracy will be selected using PSO-DBN method. Figure 5 shows the general framework of feature selection stage.

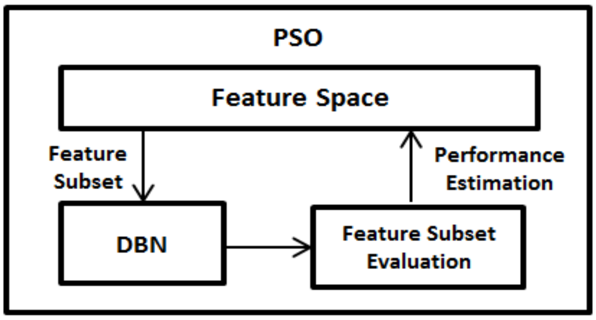


Fig. 5. General Framework of feature selection.

In the proposed system, some modifications are performed on continuous PSO version to make it suitable for feature selection task. These modifications are:

1) The step of position update in the original continuous PSO is replaced with following two steps in the proposed system (i.e., converting PSO from continues to discrete one) :

Step1: Find probability of mutation (t_{kd}) in each dimension (*d*) of each particle (*k*) using the following proposed equation:

$$t_{kd} = sign(v_{kd}) = \frac{1}{1+e^{-v_{kd}}} \tag{8}$$

Step2: Update the position of each dimension (*d*) of each particle (*k*) as follows:

$$x_{kd} = \begin{cases} P_{g,d} & if \;\; rand < t_{kd} \\ x_{kd} & if \;\; t_{kd} < rand \end{cases} \tag{9}$$

2) To prevent the stagnation problem from occurs during optimization process, an additional condition is added at the end of each iteration to check whether the problem occurs or not. If the stagnation happens then positions' values of about *pct* percentage of the total number of particles are changed randomly.

Algorithm 1 illustrates the main steps of PSO-DBN optimization method utilized by the proposed system.

Algorithm 1. PSO-DBN Optimization

Input:
N: Number of particles in the population, *MaxEpoch* : Number of iterations, *w*: Inertia factor, *c1*,*c2*: Learning rate factors, *pct*: Percentage of particles to change

Output:
Gbestposition: The position of the best forever particle
Gbestfitness: The fitness of the best forever particle

```
Initialize the particles' positions randomly with either 0 or 1
Set particles' velocities←0
Set Gbestfitness←0
Set Gbestposition←0
Begin
Set Epoch←0
While (Epoch< MaxEpoch)
        Compute fitness of each particle in the population using DBN
        Set Lbestfitness ← Fitness of best particle in the current Epoch
        Set Lbest position ← Position of best particle in the current Epoch
        If (Lbestfitness> Gbestfitness) Then
            Set Gbestfitness← Lbestfitness
            Set Gbestposition← Lbestposition
        End If
        Update particles velocities using Eq. (1)
        Update particles' positions using Eqs. (7) and (8)
        If stagnation occurs then randomly change positions of (pct ×N)
        particles
        Set Epoch←Epoch+1
End While
End
```

3.4. Classification stage

In this stage, the sign number is identified using DBN classifier. Classification stage includes training step and testing step. In training step, the DBN is trained with the training dataset for purpose of selection the proper DBN architecture and adjusting the weights which are then used in testing step to guess the number of unknown hand sign gesture.

4. Experimental Results

National University of Singapore (NUS) hand posture dataset I images are used as evaluation material. The dataset includes ten classes of postures, twenty-four hand images per class, which are captured with size 160×120 pixels. The background of hand images is uniform and the position and size of the hand differ among image samples. The hand postures are selected in such a way that the variation of interclass in the postures' appearance is small, which makes the recognition of number class is a challenge task [23]. Figure 6 shows some example images from NUS hand posture dataset.

Experiments are conducted to determine the optimal parameters' values of the proposed method. To select the optimal feature subset, PSO-DBN algorithm is applied with different *N*, *MaxEpochs*, *w*, *c1*, *c2*, and *pct* values. When *N*=10, *MaxEpochs* = 100, *w*=1, *c1* = 1, *c2*=1, *pct*=0.3 the system gives the best results, for this reason these values are adopted in the experiments.

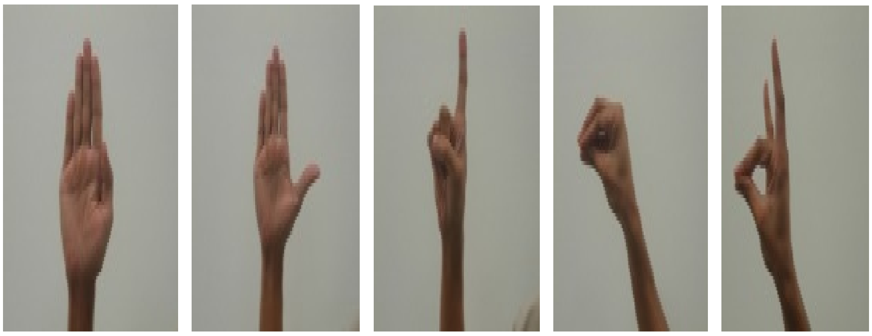


Fig. 6. Sample images from NUS hand posture dataset.

The weights of DBN are adjusted using the training part of the dataset. The purpose of training step is to determine the proper neural network configuration parameters; which are: number of nodes in hidden layer (*Hid*), learning rate (*LR*), momentum (*Mom*), and Epoch (*Ep*). Various combinations of these parameters have been tested to find the best accuracy can be reached. The accuracy is computed through dividing the number of image samples that successfully classified to one of the ten English numbers by the total number of image samples in the database. Table 1 shows the accuracy achieved to recognize the number involved within the hand image (as shown in Fig. 6) using different number of blocks along with the number of selected features by PSO-DBN and DBN parameters' settings for each case. The best configuration of DBN in each case is adopted after training and testing the DBN using different *Hid*, *LR*, *Mom* and *Ep* values. Only DBN configuration that gives the best accuracy for the given case is listed in the table.

Table 1. Accuracy achieved using different number of blocks along with DBN parameters' settings for each case.

Number of blocks	Accuracy	*T*	DBN Configuration
2	40.00 %	1	*Hid*= 5, *LR*=0.2, *Mom*=0.8, *Ep*=10000
3	56.67 %	2	*Hid*= 5, *LR*=0.1, *Mom*=0.5, *Ep*=10000
4	72.50%	5	*Hid*= 5, *LR*=0.2, *Mom*=0.7, *Ep*=10000
5	91.25%	8	*Hid*= 5, *LR*=0.2, *Mom*=0.6, *Ep*=10000
6	91.67%	8	*Hid*= 5, *LR*=0.3, *Mom*=0.6, *Ep*=10000
7	*99.58%*	*16*	*Hid= 5, LR=0.5, Mom=0.2, Ep*=10000
8	99.13%	15	*Hid*= 5, *LR*=0.5, *Mom*=0.2, *Ep*=10000

As shown in Table 1, the system gives recognition ability equal to 99.58% when the number of blocks equal to 7 blocks and number of selected features equal to 16 features which are selected from the generated Line-Slope features using PSO-DBN. Figure 7 demonstrates the fitness value of the global best particle (*Pg*) in PSO-DBN during the search for the best solution when the number of blocks equal to 7 (i.e. the total number of generated Line-Slope features= 21) while Fig. 8 shows the number of selected features in *Pg* position (i.e., the number of features which are set to one in *Pg* position vector) during optimization process.

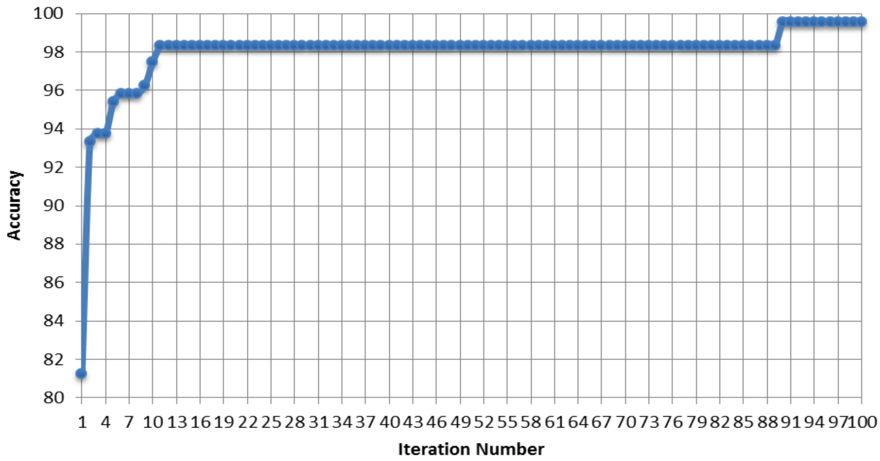


Fig. 7. ***Pg*** **fitness value during optimization process.**

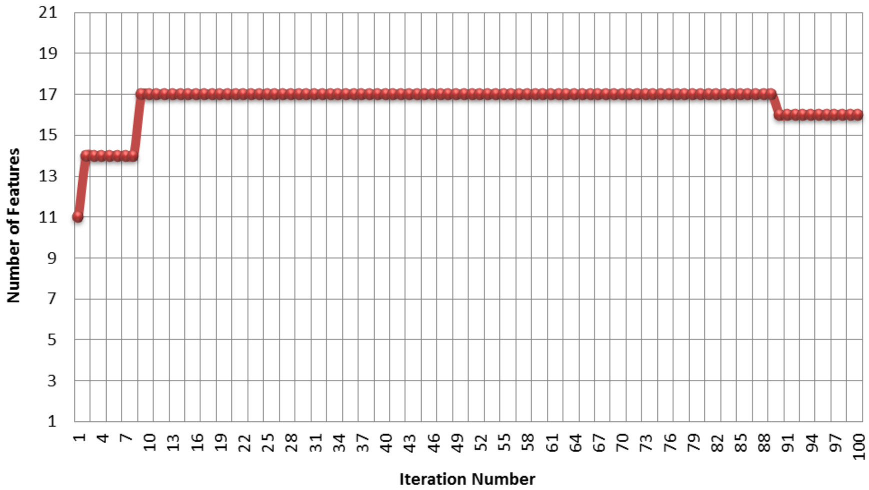


Fig. 8. Number of selected features in *Pg* position during optimization process.

As shown in Figs. 7 and 8, the accuracy increases exponentially as the total number of features increase expect for the last ten iterations in which the number of selected features decreases by one to give the best accuracy which is 99.58%.

Table 2 shows a comparison that is made with some of other recently published works on sign language recognition using NUS hand posture dataset. The authors used different methods for hand region allocation to compensate the differences in background conditions. However; the focus of our work is on feature extraction and selection problem; thus the comparison is based on the type of used features regardless of hand region allocation method. The results listed in the table indicate the outperformance of the proposed system.

Table 2. Comparison with recently published works on NUS hand posture dataset.

Authors	Feature Extraction Method	Accuracy
Nguyen et al. [12]	Kernel descriptor	97.30%
Vishwakarma [13]	Shape and texture features	94.60 %.
Ji et al. [14]	BEHB features	97.72 %
Lahiani and Neji [15]	HOG-LBP features	92.00%
The proposed system	*Line-slope features and PSO-DBN optimization method*	*99.58%*

5. Conclusions and Future Work

A system for recognizing the numbers using sign language has been proposed in this paper. HSV color model has been used for allocating hand region by thresholding saturation band value and then searching for white pixel locations in each direction (i.e., left, right, top, and bottom); the region that is enclosed by these four pixels is then considered as a hand region.

Line-slope features have been generated from the detected control points. The combination of PSO and DBN help in selection the optimal feature subset that can be used for identifying the number involved within the hand image by highlighting the effective features from the generated Line-Slope features.

The test results showed that the proposed system gave a recognition rate equal to 99.58% which reflects the effectiveness of the selected slope features. However; the proposed system may not work well with more complex background. As a future work, the hand region allocation step can be improved to allocate hand region in more complex conditions.

Corner detection algorithms such as Harris algorithm can be used to detect control point's locations. Many feature extraction methods can be employed in order to optimize the classification, for example, the edge direction within the local region of each control point can be considered as discriminated features.

Nomenclatures

B	Blue color band
c_1, c_2	Learning factors
d	Particle dimension
Ep	Number of Epochs of DBN learning
f_i	Fitness value of ith particle
G	Green color band
H	Hue color band
Hid	Number of DBN hidden nodes
k	Particle in processing
LR	DBN learning rate
M	Number of control points
$MaxEpoch$	Number of PSO-DBN iterations
$Minn$, $Maxn$	New range confines
$Mino$, $Maxo$	Old range confines
Mom	Momentum value
N	Number of particles in the population

p1, p2	Points with coordinates $(x_1,y_1),(x_2,y_2)$, respectively
Pct	Percentage of particles to change
P_g	Best forever global position vector in the population
P_l	Best local position vector in the population
R	Red color band
r_1, r_2	Random numbers between 0 and 1
S	Saturation color band
T	Number of selected features or Iteration time
t_{kd}	Probability of mutation in each dimension
V	Value color band
V_n	New feature value
V_o	Old feature value
v_i	Velocity vector of i^{th} particle
w	Inertia factor
x_i	Position vector of i^{th} particle

Abbreviations

BEHB	Binary Edge HOG Block
DBN	Deep Belief Network
HSV	Hue Saturation Value
NUS	National University of Singapore
PSO	Particle Swarm Optimization
PSO-DBN	Particle Swarm Optimization – Deep Belief Network
RGB	Red Green Blue
RBMs	Restricted Boltzmann Machines

References

1. Patel, U.; and Ambekar, A.G. (2017). Moment Based Sign Language Recognition for Indian Languages. *3rd International Conference on Computing, Communication, Control And Automation (ICCUBEA)*. Pune, India, 1-6.
2. Tolba, M.F.; and Elons, A.S. (2013). Recent Developments in Sign Language Recognition Systems. *8th International Conference on Computer Engineering & Systems (ICCES).* Cairo, Egypt, xxxvi-xlii.
3. Bantupalli, K.; and Xie, Y. (2018). American Sign Language Recognition using Deep Learning and Computer Vision. *IEEE International Conference on Big Data (Big Data)*. Seattle, WA, USA, 4896-4899.
4. Boraste, S.N.; and Mahajan, K.J. (2015). Image Processing Based Language Converter for Deaf and Dumb. *International Journal of Science, Technology & Management*, 4(5), 109-117.
5. Cooper, H.J.; Ong, N.; and Bowden, R. (2012). Sign Language Recognition Using Sub-Units. *Journal of Machine Learning Research*, 13, 2206-2231.
6. Pramada, S.D.; Saylee, N.; Pranita, N.; and Vaidya, A.S. (2013). Intelligent Sign Language Recognition Using Image Processing. *IOSR Journal of Engineering (IOSRJEN)*, 3(2), 45-51.
7. Karthikeyan, D.; and Muthulakshmi, G. (2014). English Letters Finger Spelling Sign Language Recognition System. *International Journal of Engineering Trends and Technology (IJETT)*, 10(7), 334-339.

8. Kiranalli, P.; and Gengaje, S.R. (2016). Indian Sign Language Numeral Recognition - An Image Processing Approach. *International Journal of Computer Applications*, 146(7), 24-27.
9. Kaluri, R.; and Reddy, C.P. (2016). A Framework for Sign Gesture Recognition Using Improved Genetic Algorithm and Adaptive Filter. *Cogent Engineering*, 3(1), 1-9.
10. Kaluri, R.; and Reddy, P. (2016). Sign Gesture Recognition Using Modified Region Growing Algorithm and Adaptive Genetic Fuzzy Classifier. *International Journal of Intelligent Engineering and Systems*, 9(4), 225-233.
11. Kaluri, R., and Pradeep, C.H. (2017). An Enhanced Framework for Sign Gesture Recognition Using Hidden Markov Model and Adaptive Histogram Technique. *International Journal of Intelligent Engineering and Systems*, 10(3), 11-19.
12. Nguyen, V.; Le, T.; Tran, T.; Mullot, R.; and Courboulay, V. (2014). Hand Posture Recognition Using Kernel Descriptor. *Procedia Computer Science*, 39,154-157.
13. Vishwakarma, D.K. (2017). Hand Gesture Recognition Using Shape and Texture evidences in Complex Background. *Proceedings of the International Conference on Inventive Computing and Informatics (ICICI 2017)*. Coimbatore, India, 278-283.
14. Ji, P.; Song, A.; and Xiong, P. (2017). Egocentric-Vision Based Hand Posture Control System for Reconnaissance Robots. *Journal of Intelligent & Robotic Systems*, 87 (3), 583–599.
15. Lahiani, H.; and Neji, M. (2018). Hand Gesture Recognition Method Based on HOG-LBP Features for Mobile Devices. *Procedia Computer Science*, 128, 254–263.
16. Dai, Y.; Liu, L.; and Li, Y. (2011). An Intelligent Parameter Selection Method for Particle Swarm Optimization. *Fourth International Joint Conference on Computational Sciences and Optimization*. Yunnan, China, 960-964.
17. Bondarenko, A.; and Borisov, A. (2013). Research on the Classification Ability of Deep Belief Networks on Small and Medium Datasets. *Information Technology and Management Science*, 16, 60-65.
18. Pezeshki, M.; and Gholami, S. (2014). Distinction Between Features Extracted Using Deep Belief Networks. Retrieved July 2, 2019, from https://arxiv.org/abs/1312.6157.
19. Suhaila, N.M; Jabir, A.J., and Abbas. Z.A. (2019). Spin-Image Descriptors for Text-Independent Speaker Recognition *In: Saeed, F.; Mohammed, F.; and Gazem, N. (eds) Emerging Trends in Intelligent Computing and Informatics. IRICT 2019. Advances in Intelligent Systems and Computing,* 1073, Springer, Cham, 216-226
20. Gonzalez, R.C.; and Woods, R.E. (2002). *Digital Image Processing* (3rd ed.). United States: Pearson Education.
21. Weir, M.D.; Hass, J.; and Giordano, F.R. (2008). *Thomas Calculus*: Early Transcendentals (11th ed.). United States: Addison-Wesley Longman.
22. Witten, I.H.; and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques* (2nd ed.). San Francisco: Elsevier.
23. The NUS hand Posture Dataset Home Page. Retrieved August 10, 2019, from www.ece.nus.edu.sg/stfpage/elepv/NUS-HandSet.